SDI-12 PROTOCOL
FOR TRIME- PICO
DEVELOPER'S MANUAL

DATE: 2012-12-03, VERSION-1.01

IMKO GMBH
IM STOECK 2
D-76275 ETTLINGEN

PHONE: +49-(0)7243-5921-17
FAX: +49-(0)7243-90856

E-MAIL: INFO@IMKO.DE

TABLE OF CONTENTS

1 INTRODUCTION

SDI-12 is a standard for interfacing data recorders with microprocessor-based sensors. SDI-12 stands for serial/digital interface at 1200 baud rate.

1.1 SDI-12 Electrical Interface

The SDI-12 electrical interface uses the SDI-12 bus to transmit serial data between SDI-12 data recorders and sensors. The SDI-12 bus is the cable that connects multiple SDI-12 devices.
This is a cable with three conductors:
1) a serial data line
2) a ground line
3) a 12-volt line

1.2 Baud Rate and Byte Frame Format

The baud rate for SDI-12 is 1200. The following shows the byte frame format for SDI-12.
• 1 start bit
• 7 data bits, least significant bit transmitted first
• 1 parity bit, even parity
• 1 stop bit

1.3 SDI-12 Communications Protocol

SDI-12 data recorders and sensors communicate by an exchange of ASCII characters on the data line. The data recorder sends a break to wake up the sensors on the data line. A break is continuous spacing on the data line for at least 12 milliseconds. The data recorder then sends a command. The sensor, in turn, returns the appropriate response. Each command is for a specific sensor. The first character of each command is a unique sensor address that specifies with which sensor the recorder wants to communicate. Other sensors on the SDI-12 bus ignore the command and return to low-power standby mode. When a data recorder tells a sensor to start its measurement procedure, the recorder does not communicate with any other sensor until the data collection from the first sensor is complete. (During a concurrent measurement command, however, a data recorder can communicate with other sensors while one or more sensors are taking measurements.)

1.4 Device Addresses

The first character of every command must be a sensor address. Likewise, the first character of a response is also the address character. This lets an SDI-12 recorder verify that the response has come from the correct sensor. (An address is a single character used to indicate which sensor is to respond to the command.
ASCII '0' (decimal 48) through ASCII '9' (decimal 57) are the standard addresses which all sensors and data recorders must support. Should there be a need for more

than 10 sensors, use an address in the range ASCII 'A' through ASCII 'Z' (decimal 65 through 90) and ASCII 'a' through ASCII 'z'(decimal 97 through 122).

2 SDI-12 COMMANDS AND RESPONSES

Refer to the document "SDI-12 **A Serial-Digital Interface Standard for Microprocessor-Based Sensors**." Version 1.3 January 12, 2009 Prepared by the SDI-12 Support Group.

During normal communication, the data recorder sends an address together with a command to a Trime-Pico SDI-12 sensor. The Trime-Pico replies with a "response". In the following descriptions, SDI-12 commands and responses are enclosed in quotes. The SDI-12 address and the command/response terminators are defined as follows:

"a" Is the sensor address. The following ASCII Characters are valid addresses: "0-9", "A-Z", "a-z". Sensors will be initially programmed at the factory with the address of "0" for use in single sensor systems. Addresses "1 to 9" and "A to Z" or "a to z" can be used for additional sensors connected to the same SDI-12 bus. Address "*" and "?" are "wild card" addresses which select any sensor, regardless of its actual address.

"!" Is the last character of a command block.

<CR><LF>" Are carriage return (0D) hex and line feed (0A) hex characters. They are the last two characters of a response block.

Notes:

- All commands/responses are upper-case printable ASCII characters.
- Commands must be terminated with a "!" character.
- Responses are terminated with <CR><LF>characters.
- The command string must be transmitted in a contiguous block with no gaps of more than 1.66 milliseconds between characters.

2.1 Master SDI-12 Command List

Standard SDI-12 commands

Command	Description
a!	Acknowledge Active
aI!	Send Identification
aAb!	Change Address
?!	Address Query
aM!	Start Measurement
aMC!	Start Measurement and Request CRC
aD0!	Send Data
aV!	Start Verification
aC!	Start Concurrent Measurement
aCC!	Start Concurrent Measurement and Request CRC

Extended SDI-12 commands unique to the Trime-Pico

Command	Description
aXGP!	Get the Protocol
aXSP+p!	Set the Protocol(p-0:IMP, p-1:SDI-12)

aXGC!	Get the calibration
aXSC+cc!	Set the calibration(cc: 01--15)

2.2 Acknowledge Active Command(a!)

The Acknowledge Active Command returns a simple status response which includes the address of the sensor. Any measurement data in the sensor's buffer is not disturbed.

Command	Response
"a!"	"a<CR><LF>"

Where: a is the sensor address ("0-9", "A-Z", "a-z").

Example:

Command	Response
0!	0<CR><LF>

2.3 Send Identification Command(aI!)

The Send Identification Command responds with sensor vendor, model, firmware and hardware version.

Command	Response
"aI!"	"allccccccccmmmmmmvvvxx...xx<CR><LF>"

Where:

a	is the sensor address ("0-9", "A-Z", "a-z").
I	is an upper-case ASCII character.
ll	is the SDI-12 version compatibility level, e.g. version 1.3 is represented as "13".
cccccccc	is an 8 character vendor identification to be specified by the vendor and usually in the form of a company name or its abbreviation.
mmmmmm	is a 6 character field specifying the sensor model number.
vvv	is a 3 character field specifying the sensor firmware version number.
xx...xx	is an optional field of up to a maximum of 13 characters to be used for serial number and hardware version number.

Example:

Command	Response
0I!	013IMKOGmbHPico3200135001-1.14<CR><LF>

Where:

0: sensor address
13: SDI-12 version 1.3
IMKOGmbH: company name
Pico32:sensor model
001: sensor firmware version

35001: sensor serial number
1.14: sensor hardware version

2.4 Change Address Command(aAb!)

The Change Address Command allows the sensor address to be changed.

Command	Response
"aAb!"	"b<CR><LF>"

Where:

- a is the current (old) sensor address ("0-9", "A-Z", "a-z").
- A is an upper-case ASCII character.
- b is the new sensor address to be programmed ("0-9", "A-Z","a-z").

Example:

Command	Response
1A2!	2<CR><LF>

Where:

- 1 is the old sensor address.
- A stands for the Change Address command.
- 2 is the new sensor address.

2.5 Address Query Command(?!)

The command allows to query the address of the sensor if only one sensor is connected on the bus.

Command	Response
"?!"	"a<CR><LF>"

Where:

- a is the sensor address ("0-9", "A-Z", "a-z").

Example:

Command	Response
?!	1<CR><LF>

2.6 Start Measurement Command(aM!)

This command tells the sensor to take a measurement. The sensor does not return the measurement to the data recorder after this command. It returns the time until the measurements will be ready and the number of measurements that it will make. The send data (D0!) command must be issued to get the measurement(s).

Command	Response
"aM!"	"atttn<CR><LF>"

Where:

a	is the sensor address ("0-9", "A-Z", "a-z").
M	stands for starting measurement.
ttt	is a three digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data.
n	is the number of measurement values the sensor will make and return in one subsequent D command, a single digit integer from 0 to 9.

Upon completion of the measurement, a service request "a<CR><LF>" is sent to the data recorder indicating the sensor data is ready. The data recorder may wake the sensor with a break and collect the data any time after the service request is received or the specified processing time has elapsed.

Example:

Command	Response
1M!	10053<CR><LF>

Where:

1	is the sensor address.
005	is the maximum time, 5 seconds to complete the measurement.
3	means 3 measurement values.

Subsequent Command	Response
1D0!	1+13.24+25.00+20.00<CR><LF>

Where:

1	the sensor address.
D0	the Send Data command.
13.24	the moisture value(%).
25.00	the temperature value(°C). If applicable, refer to the sensor user manual.
20.00	the Trime-EC value(S/m). If applicable, refer to the sensor user manual.

2.7 Start Measurement and Request CRC Command(aMC!)

The command is the same as the command aM! except that it requests the data be returned with a 16 bit Cyclic Redundancy Check (CRC) appended to it.

Command	Response
"aMC!"	"atttn<CR><LF>"

Where:

a	is the sensor address ("0-9", "A-Z", "a-z").
MC	stands for starting measurement with CRC.
ttt	is a three digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data.
n	is the number of measurement values the sensor will make and return

in one subsequent D command, a single digit integer from 0 to 9.

Upon completion of the measurement, a service request "a<CR><LF>" is sent to the data recorder indicating the sensor data is ready. The data recorder may wake the sensor with a break and collect the data any time after the service request is received or the specified processing time has elapsed.

Example:

Command	Response
1M!	10053<CR><LF>

Where:

1	is the sensor address.
005	is the maximum time, 5 seconds to complete the measurement.
3	means 3 measurement values.

Subsequent Command	Response
1D0!	1+13.24+25.00+20.00KOj<CR><LF>

Where:

1	the sensor address.
D0	the Send Data command.
13.24	the moisture value(%).
25.00	the temperature value(°C). If applicable, refer to the sensor user manual.
20.00	the Trime-EC value(S/m). If applicable, refer to the sensor user manual.
KOj	CRC of 1+13.24+25.00+20.00.

2.8 Send Data Command(aD0!)

This command is used to get groups of data from the sensor. D0! is issued after an M, MC, C, CC, or V command. The sensor responds by sending the data. It will be used also in the extended commands.

Command	Response
"aD0!"	"a<values><CR><LF>" or "a<values><CRC><CR><LF>"

Where:

a	is the sensor address ("0-9", "A-Z", "a-z").
D0	the Send Data command.
<values>	pd.d p - the polarity sign (+ or -) d - numeric digits before the decimal place . - the decimal point (optional) d - numeric digits after the decimal point
<CRC>	3 character CRC code, appended if data was requested with CRC

The command is not an independent one. It depends on which command is carried out prior to it. So its examples are in other commands.

2.9 Start Verification(aV!)

This command tells the sensor to return the system error and the application error in response to a subsequent D command. The format of this command is the same as the M commands. The format of the response is the same as the D commands.

Command	Response
"aV!"	"atttn<CR><LF>"

Where:

a	is the sensor address ("0-9", "A-Z", "a-z").
V	stands for starting verification.
ttt	is a three digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data.
n	is the number of measurement values the sensor will make and return in one subsequent D command, a single digit integer from 0 to 9.

Example:

Command	Response
1V!	10002<CR><LF>

Where:

1	is the sensor address.
000	waits no time, 0 second.
2	means 2 values.

Subsequent Command	Response
1D0!	1+000+000<CR><LF>

Where:

1	the sensor address.
D0	the Send Data command.
000	no error, the system error.
000	no error, the application error.

2.10 Start Concurrent Measurement Command(aC!)

This command tells the sensor to take a concurrent measurement. A concurrent measurement is one which occurs while other SDI-12 sensors on the bus are also taking measurements. This command is similar to the “aM!” command, however, the nn field has an extra digit and the sensor does not issue a service request when it has completed the measurement. Communicating with other sensors will NOT abort a concurrent measurement. The send data (D0!) command must be issued to collect the measurements(s).

Command	Response
"aC!"	"atttnn<CR><LF>"

Where:

a is the sensor address ("0-9", "A-Z", "a-z").
C stands for starting concurrent measurement.
ttt is a three digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data.
nn is the number of measurement values the sensor will make and return in one subsequent D command.

Example:

Command	Response
1C!	100503<CR><LF>

Where:

1 is the sensor address.
005 is the maximum time, 5 seconds to complete the measurement.
3 means 3 measurement values.

Subsequent Command	Response
1D0!	1+13.24+25.00+20.00<CR><LF>

Where:

1 the sensor address.
D0 the Send Data command.
13.24 the moisture value(%).
25.00 the temperature value(°C). If applicable, refer to the sensor user manual.
20.00 the Trime-EC value(S/m). If applicable, refer to the sensor user manual.

2.11 Start Concurrent Measurement and Request CRC Command(aCC!)

The command is the same as the command aC! except that it requests the data be returned with a 16 bit Cyclic Redundancy Check (CRC) appended to it.

Command	Response
"aCC!"	"atttnn<CR><LF>"

Where:

a is the sensor address ("0-9", "A-Z", "a-z").
CC stands for starting concurrent measurement with CRC.
ttt is a three digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data.
nn is the number of measurement values the sensor will make and return in one subsequent D command.

Upon completion of the measurement, a service request "a<CR><LF>" is sent to the data recorder indicating the sensor data is ready. The data recorder may wake the sensor with a break and collect the data any time after the service request is received or the specified processing time has elapsed.

Example:

Command	Response
1CC!	10053<CR><LF>

Where:

1	is the sensor address.
005	is the maximum time, 5 seconds to complete the measurement.
3	means 3 measurement values.

Subsequent Command	Response
1D0!	1+13.24+25.00+20.00KOj<CR><LF>

Where:

1	the sensor address.
D0	the Send Data command.
13.24	the moisture value(%).
25.00	the temperature value(°C). If applicable, refer to the sensor user manual.
20.00	the Trime-EC value(S/m). If applicable, refer to the sensor user manual.
KOj	CRC of 1+13.24+25.00+20.00.

3 APPENDIX A ERROR MESSAGE

The following table lists all error messages.

Error Message Table Version 1.00

Code	Error Message
0	no error
	System Errors(1--99)
	Serial communication errors
1	received address block invalid, first byte should be 253
2	received address block check CRC
3	received data block check CRC
4	time out of receiving data block
5	V24
6	UART
7	time out of transmitting address block
8	time out of transmitting data block
	Command errors
20	no command
21	parameter number isn't in table
22	get page parameter
23	set page parameter
24	parameter is not writable
25	TDR scan parameter
26	have no support right
27	setting all parameters needs support right
28	the event of TDRScan must be set in advance
29	baud rate is too small
30	baud rate is too big
	EEPROM errors
41	no response from EEPROM
42	page writing is out of range
43	SCL stuck to low
44	SDA stuck to low
45	write memory address
46	write data to EEPROM
47	get & set image parameter
	ASIC errors
50	ASIC check is failed
	Others
60	Voltage too low

	Application Errors(100--199)
	TDR measuring errors
100	TDR position is over flow
101	TDR position is under flow
102	get ASIC temperature
103	EC divisor is zero
105	Tp is out of range
106	Resister is out of range
108	No reflect point is found
	ADC,DAC, material temperature
120	A/D convert
121	D/A convert
122,123	get material temperature
	Calculate
130	minimal or maximal gain threshold in calculating coefficients
131	divisor is zero in ASIC temperature compensation
132	actual moisture is too large in DAC
133	actual moisture is too small in DAC
134	actual temperature is too large in DAC
135	actual temperature is too small in DAC
136	TpMDivisor is zero
137	T to Ms mode is out of range
138	Ratio divisor is zero
	SelfTest
150	DAC code is too small
151	DAC code is too large
200-254	reserved
255	more data